A133 572

(12)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) UNCLASSIFIED

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DYNAMIC CONTROL OF SESSION INPUT RATES IN COMMUNICATION NETWORKS		5. TYPE OF REPORT & PERIOD COVERED Technical 6. PERFORMING ORG. REPORT NUMBER LIDS-P-1298
7. AUTHOR(s) Eli Gafni and Dimitri Bertsekas		8. CONTRACT OR GRANT NUMBER(s) ARPA Order No. 3045/5-7-75 ONR/N00014-75-C-1183
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology Laboratory for Information and Decision Systems Cambridge, Massachusetts 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Code No. 5T10 ONR Identifying No. 049-383
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		12. REPORT DATE May 1983 13. NUMBER OF PAGES 27
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We consider a distributed algorithm for dynamically adjusting the input rate of each session of a voice or data network so as to exercise flow control. Each session origin receives periodically information regarding the level of congestion along the session path and iteratively corrects its input rate. In this paper we place emphasis on voice networks but the ideas involved are also relevant for dynamic routing and flow control in data networks.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

May 1983

LIDS-P-1298

DYNAMIC CONTROL OF SESSION INPUT RATES IN COMMUNICATION NETWORKS*

by

Eli Gafni† and Dimitri Bertsekas††

Abstract

The authors ~~We~~ consider a distributed algorithm for dynamically adjusting the input rate of each session of a voice or data network so as to exercise flow control. Each session origin receives periodically information regarding the level of congestion along the session path and iteratively corrects its input rate. In this paper ~~we~~ they place emphasis on voice networks but the ideas involved are also relevant for dynamic routing and flow control in data networks.

*This research was conducted at the M.I.T. Laboratory for Information and Decision Systems with partial support provided by the Defense Advanced Research Projects Agency under Contract No. ONR/N00014-75-C-1183.

†The author is with the University of California at Los Angeles, Computer Science Dept., Los Angeles, California 90024.

††The author is with the Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, Cambridge, MA 02139.

1. INTRODUCTION

The purpose of this paper is to propose and investigate a flow control algorithm for adjusting session rates in a data or voice network. This algorithm is motivated by a Voice Coder scheme introduced in [1] and called "embedded coding". In this scheme a segment of talkspurt is coded into packets of different priority levels. The higher priority packets contain the "core" of the speech while the lower priority packets contain the information that "fine tunes" it. Traditional voice flow control mechanisms either block the initiation of a call or discard small segments of it while it is already in progress. By contrast the embedded coding scheme dynamically trades off between voice quality and congestion by discarding the lower "priority" packets either at the point of congestion or the point of entry. The level of congestion at which the gaps between the segments, delivered by the traditional schemes, render the speech unintelligible is much lower than the one at which the embedded coding scheme delivers unintelligible information. This flexibility in exercising flow control makes the embedded coding scheme attractive.

Alleviation and prevention of congestion by discarding lower priority packets at the point of entry seems to be superior to discarding them at the point of congestion. The latter amounts to a waste of network resources. But, it would not be advisable to forgo the capability of discarding lower priority packets at the point of congestion, because of the time delay involved in making the entry points aware of downstream congestion. Based on this we believe that both capabilities should be used. The rates at

the entry points will be determined upon longer time averages of congestion levels while the capability of discarding packets at the point of congestion will serve to alleviate intolerable momentary bottlenecks. The rates at the entry point will be adjusted so that the capability of discarding packets at the point of congestion will not be exercised too often.

In this paper we discuss a method of determining the input rates at the entry point. To this end we will ignore the capability of discarding packets within the network in order to simplify the analysis. As in quasistatic routing we employ an "on-line" iterative algorithm that will solve a static problem. The hope is that the algorithm converges fast enough relative to the session initiation and termination process, and as a result will be able to "track" its variation keeping the rates in the ballpark of the optimal rates at all times.

The criterion used to determine input rates is based on the notion of "fair allocation" introduced in Section 2. Roughly speaking the objective is to maximize the smallest session rate, and once this is achieved to maximize the second smallest rate, etc. In Section 3 we introduce the algorithm and describe its convergence properties.

The idea of the algorithm is to adjust the input rates of sessions on the basis of the current level of congestion along the session path. The necessary information is collected by a control packet sent periodically by each session origin along the path similarly as in flow control methods investigated by simulation in [1]. This method of adjusting input rates seems also suitable for other situations where fast reaction to momentary congestion is needed. For example when the number of users

in the network is small but some of these users can overload the network if left uncontrolled to transmit at maximum rate, then a dynamic method of routing and flow control is needed. The ideas of this paper can provide an alternative to existing techniques [7], [8] in such situations. Further work is required along this direction.

2. PROBLEM FORMULATION

Consider a network with nodes 1, 2, ..., N and a set of directed links L. Each link $a \in L$ has a capacity C_a associated with it--a positive number. Let S denote a set of sessions taking place between nodes. Each session $s \in S$ has an origin node associated with it and traverses a subset of links denoted by L_s. Note that we do not restrict the session to have a single destination, so the set of links L_s may be for example a tree rooted at the origin node of s and used for broadcasting messages throughout the network. We denote by S_a the set of sessions traversing a link $a \in L$. If r_s is the input rate of session s (in data units/sec), then the flow F_a of a link $a \in L$ is given by

$$F_a = \sum_{s \in S_a} r_s. \tag{1}$$

The problem broadly stated is to choose a vector of session input rates $r = (..., r_s, ...)$ which results in a set of "satisfactory" link flows $\{F_a | a \in L\}$, and at the same time maintains a certain degree of "fairness" for all sessions.

It is customary to consider as one of the characteristics of a fair allocation of resources in a network the feature that it is indifferent to the geographical separation of the session's origin and destinations. Although there might be different priorities assigned to sessions, these priorities are not assigned on the basis of geographical distance. Moreover, two sessions of the same priority should obtain the same rate, if the rate of one can be traded for the rate of the other without overloading the network or reducing the rate of any other session. This is in the spirit of making the network "transparent" to the user.

To capture the notion of fairness and priority as presented above we define the notion of fair allocation:

For a vector $x = (x^1, x^2, \ldots, x^n)$ in the Euclidean space R^n, we consider the vector $\bar{x} = (\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^n)$ the coordinates of which are the same as those of x but are rearranged in order of increasing value, i.e. we have $\bar{x}^1 \leq \bar{x}^2 \ldots \leq \bar{x}^n$ and with each $i = i, \ldots, n$ we can associate a distinct i' such that $\bar{x}^i = x^{i'}$. We call $\bar{x}$ the <u>increasing permutation</u> of x. Given a subset X of R^n we will say that a vector $x \in X$ is a <u>fair allocation</u> over X if for every vector $y \in X$ the increasing permutation $\bar{x}$ of x is lexicographically greater or equal to the increasing permutation $\bar{y}$ of y, i.e. if $\bar{y}^j > \bar{x}^j$ for some j, then there exists an $i < j$ such that $\bar{y}^i < \bar{x}^i$.

If we view X as a "feasible" set, a fair allocation vector x over X solves a hierarchy of problems. The first problem is to maximize the minimal coordinate of vectors in X. The second problem is to maximize the second minimal coordinate over all vectors which solve the first problem, etc.

Hayden [2] proposed an algorithm which results in a rate vector $r = (\ldots, r_s, \ldots)$ which is a fair allocation over the set defined by

$$F_a \leq \rho C_a, \qquad \forall\ a \in L, \tag{2}$$

where ρ is some constant between 0 and 1. Jaffe [3] proposed an algorithm which obtains a rate vector r such that the vector $(\ldots, \beta_s r_s, \ldots)$ is a fair allocation over the set defined by

$$\beta_s r_s \leq C_a - F_a, \qquad \forall\ s \in S,\ a \in L_s, \tag{3}$$

$$F_a \leq C_a, \quad \forall\ a \in L, \qquad r_s \geq 0, \quad \forall\ s \in S, \tag{4}$$

where β_s is some positive constant that characterizes the priority of session s.

The rationale behind the fair allocation problem based on (2) is quite simple: we maximize the minimum session rate while not allowing the flow of any link to be more than some given fraction of its capacity. The rationale behind (3), (4) is somewhat more sophisticated. Primarily it enables us to establish preferences among sessions, and to accomodate fluctuations of a session rate which depend linearly on the rate as we will demonstrate shortly.

While Jaffe's algorithm is not iterative and as a result is somewhat unsuitable for distributed operation, Hayden's algorithm may result in transient flows that are larger than some link capacities (for an example see [4], p. 39).

Our purpose in this paper is to propose and analyze an iterative algorithm that solves a problem that is more general than Jaffe's [3], maintains at all times feasibility of link flows with respect to capacities, and is suitable for distributed operation. To this end we generalize the set defined by (3), (4) as follows:

For each link $a \in L$ and session $s \in S$ let $g_a: R^+ \to R^+$ and $\beta_s: R^+ \to R^+$ be functions mapping the nonnegative portion of the real line R^+ into itself. We are interested in finding a rate vector r such that the vector $(\ldots,\beta_s(r_s),\ldots)$ is a fair allocation over the set defined by

$$\beta_s(r_s) \leq g_a(C_a - F_a), \qquad \forall\ s \in S,\ a \in L_s \tag{5}$$

$$F_a \leq C_a, \qquad \forall\ a \in L,\ r_s \geq 0,\ \forall\ s \in S. \tag{6}$$

A vector r with this property will be called a fair allocation rate.

We make the following assumptions regarding the functions $g_a(\cdot)$ and $\beta_s(\cdot)$:

Assumption A: For all $a \in L$, $g_a(\cdot)$ is monotonically nondecreasing, and, for all $s \in S$, $\beta_s(\cdot)$ is continuous, monotonically increasing, and maps R^+ onto R^+. (This implies also that the inverse $\beta_s^{-1}(\cdot)$ exists, is continuous, monotonically increasing and maps R^+ onto R^+).

Assumption B: The function $H_{sa}(\cdot)$ defined by

$$H_{sa}(f) = \beta_s^{-1}[g_a(f)], \qquad \forall\, s \in S,\ a \in L,\ f \in R^+$$

is convex and differentiable on R^+ and satisfies

$$H_{sa}(0) = 0.$$

Assumption B is not very restrictive. It is satisfied in particular if both $\beta_s^{-1}(\cdot)$ and $g_a(\cdot)$ are convex, differentiable and monotonically increasing on R^+, and $g_a(0) = 0$. Also the convexity assumption in Assumption B can be replaced by a concavity assumption without affecting the convergence result of the next section, but this will not be pursued further.

The introduction of the nonlinear function $\beta_s(\cdot)$ allows us to assign different priorities to different sessions in a more flexible manner than in (3), and allows additional freedom in mathematically expressing algorithmic design objectives. As an example let us provide justification for the use of a particular form for g_a in the case where each session is a voice conversation.

Suppose that the length of time over which each session rate is averaged is short relative to the "time constant" of the counting process of the number of off-hook speakers which are currently in talkspurt mode. Since about 30% of a talkspurt is silence and some segments of the talkspurt need more encoding than others, we view the bit rate generated by the Vocoder for session $s \in S$ as a stochastic process with mean r_s--the rate assigned to user $s \in S$. We thus implicitly assume that the Vocoder has the means of dynamically reconfiguring to the demands of the voice to achieve the desired average rate. Suppose that we want to reserve excess capacity on each link so as to be able to accomodate a variation at least as large as the standard deviation of the flow on the link. Assume that the standard deviation of the rate of each session $s \in S$ is $\gamma \cdot r_s$ where $0 < \gamma < 1$. For a fixed link $a \in L$ let $s' \in S$ be such that

$$s' = \arg \max_{s \in S_a} r_s .$$

Then, by the independence of the rates of different sessions, we have, assuming $F_a \leq C_a$, that the standard deviation $\sigma(F_a)$ of the flow F_a satisfies

$$\sigma(F_a) = \sqrt{\sum_{s \in S_a} [\sigma(r_s)]^2} = \gamma \sqrt{\sum_{s \in S_a} r_s^2}$$

$$\leq \gamma \sqrt{(\sum_{s \in S_a} r_s) r_{s'}} \leq \gamma \sqrt{C_a r_{s'}}$$

Suppose we take in (5)

$$\beta_{s'}(r_{s'}) = r_{s'}, \quad g_a(C_a - F_a) = \frac{1}{\gamma^2 C_a}(C_a - F_a)^2 \tag{8}$$

Then from (5), (7) and (8) we obtain

$$\sigma(F_a) \leq \gamma\sqrt{C_a r_{s'}} \leq \gamma\sqrt{C_a g_a(C_a - F_a)} = C_a - F_a.$$

We are thus guaranteed to be able to accomodate the standard deviation of the flow resulting from the fair allocation.

In the second interpretation, the length of time over which the rate is averaged is relatively long with respect to the "time constant" of the counting process of the number of off-hook speakers in talkspurt mode. In this case we deal concurrently with all the off-hook sessions and want to be able to accommodate the standard deviation around the mean of the process (i.e., the instantaneous effect of the number of speakers at the talkspurt mode is washed out by the long time average). Let q be the fraction of time a speaker is in the talkspurt mode and assume his rate while in the talkspurt mode is constant. Then using notations as before

$$\sigma(F_a) = \left[\sum_{s\in S_a} (r_s/q)^2 \cdot q(1-q)\right]^{1/2}$$

$$\leq \left(\frac{1-q}{q} C_a r'_s\right)^{1/2}$$

$$\leq \left(\frac{1-q}{q} C_a\right)^{1/2} [g_a(C_a - F_a)]^{1/2}.$$

Again, by choosing g_a as in (8) with $\gamma = (\frac{q}{1-q})^{1/2}$ we obtain $\sigma(F_a) \leq C_a - F_a$.

The point we want to make by the above arguments is that there is often a need to allow g_a to be a nonlinear function, which may depend also on C_a, rather than only on the excess capacity as (3) implies. The exact role of g_a is up to the network designer to decide, and our formulation allows him a great deal of flexibility in this regard.

It is possible to show that Assumptions A and B guarantee existence and uniqueness of a fair allocation rate. The proof given below is constructive and is based on a finitely terminating algorithm. However this algorithm, in contrast with the one of the next section, is not suitable for distributed, on-line operation since it must be restarted each time an old session is terminated or a new one is initiated.

Consider first the problem of finding a vector $r = (\ldots, r_s, \ldots)$ that maximizes

$$\min_{s \in S} \beta_s(r_s)$$

over the feasible set

$$R_0 = \{r \mid (5) \text{ and } (6) \text{ are satisfied}\}$$

This is the first problem in the hierarchy of problems solved by a fair allocation rate, and can be solved simply by observing that its optimal value [i.e. $\max_{r \in R_0} \min_{s \in S} \beta_s(r_s)$] is equal to

$$w_1^* = \max\{w \mid w \leq g_a[C_a - \sum_{s \in S_a} \beta_s^{-1}(w)],\ a \in L\} \ . \qquad (9)$$

This follows easily from the fact that both g_a and β_s^{-1} are monotonically nondecreasing. Denote

$$L^*(1) = \{a \in L \mid w_1^* = g_a [C_a - \sum_{s \in S_a} \beta_s^{-1}(w_1^*)]\}$$

$$S^*(1) = \{s \in S \mid L_s \cap L^*(1) \neq \emptyset\}.$$

For any fair allocation rate $(\ldots, r_s, \ldots)$ the rate of the sessions in $S^*(1)$ is equal to $\beta_s^{-1}(w_1^*)$, i.e.

$$r_s = \beta_s^{-1}(w_1^*), \qquad \forall\, s \in S^*(1),$$

while $L^*(1)$ may be viewed as the set of bottleneck links the presence of which does not allow us to increase $\min_{s \in S} \beta_s(r_s)$ beyond the level w_1^*. Therefore for the purposes of determining further a fair allocation rate vector, the rates of the sessions $s \in S^*(1)$ are fixed at $\beta_s^{-1}(w_1^*)$ and we can consider a reduced network whereby the links $a \in L^*(1)$ and sessions $s \in S^*(1)$ are eliminated while the capacity C_a of each link $a \notin L^*(1)$ is replaced by

$$C_a - \sum_{s \in S^*(1) \cap S_a} \beta_s^{-1}(w_1^*).$$

If $S^*(1) = S$ we are done; otherwise we can consider the problem of maximizing the minimal value of $\beta_s(r_s)$ in the reduced network similarly as earlier. This will determine a new optimal value w_2^* with $w_2^* > w_1^*$, a

a new set of bottleneck links $L^*(2)$, and a set of sessions $S^*(2)$ such that

$$\beta_s(r_s) = w_2^*, \quad \forall\, s\epsilon S^*(2)$$

in any fair allocation vector. If $S^*(1)\cup S^*(2) = S$ we are done; otherwise we can proceed by constructing a reduced network and continue in the same manner as earlier until we exhaust all sessions. This argument constructs a fair allocation rate r^* and shows that it is uniquely defined in terms of the scalars $w_1^*, w_2^*,\ldots$, and the corresponding sets $S^*(1), S^*(2),\ldots$ Note also that the session rates r_s^*, $s\epsilon S$ and associated link flows $F_a^*, a\epsilon L$ satisfies

$$r_s^* = \min_{a\epsilon L_s} H_{sa}(C_a - F_a^*), \quad \forall\, s\epsilon S. \tag{10}$$

The algorithm of the next section is based on this property.

We can also show the reverse property namely that if a rate vector r^* satisfies (10) then it is a fair allocation rate. To see this let $\tilde{r} = (\ldots,\tilde{r}_s,\ldots)$ satisfy (10) or equivalently

$$\beta_s(\tilde{r}_s) = \min_{a\epsilon L_s} g_a(C_a - \tilde{F}_a), \quad \forall\, s\epsilon S. \tag{11}$$

Observe that from the definition (9) of w_1^* and (11) we obtain

$$w_1^* \geq \tilde{w}_1$$

where

$$\tilde{w}_1 = \min_{s\epsilon S} \min_{a\epsilon L_s} g_a(C_a - \tilde{F}_a).$$

Let $\tilde{a} \in \mathcal{L}$ be any link such that

$$g_{\tilde{a}}(C_{\tilde{a}} - \tilde{F}_{\tilde{a}}) = \tilde{w}_1.$$

Then from (11) we have

$$\tilde{r}_s = \beta_s^{-1}(\tilde{w}_1), \quad \forall\, s \in \mathcal{S}_{\tilde{a}}.$$

The inequality $\tilde{w}_1 \leq w_1^*$ implies that $\tilde{F}_{\tilde{a}} = \sum_{s \in \mathcal{S}_{\tilde{a}}} \tilde{r}_s \leq \sum_{s \in \mathcal{S}_{\tilde{a}}} r_s^* = F_{\tilde{a}}^*$ where r^* is the fair allocation rate. But this implies that

$$\tilde{w}_1 = g_{\tilde{a}}(C_{\tilde{a}} - \tilde{F}_{\tilde{a}}) \geq g_{\tilde{a}}(C_{\tilde{a}} - F_{\tilde{a}}^*) \geq w_1^*.$$

Therefore we must have $\tilde{w}_1 = w_1^*$ and it follows that the vector $\tilde{r}$ solves the first problem in the hierarchy of the fair allocation problem. Proceeding similarly as earlier we can show that $\tilde{r}$ solves all the problems in the hierarchy of the fair allocation problem and is therefore a fair allocation rate.

We summarize the conclusions from the preceding arguments in the following proposition.

Proposition 1: Let Assumptions A and B hold.

a) There exists a unique fair allocation rate.

b) $r^* = (\ldots, r_s^*, \ldots)$ is a fair allocation rate if and only if it satisfies

$$\beta_s(r_s^*) = \min_{a \in \mathcal{L}_s} g_a(C_a - F_a^*), \quad \forall\, s \in \mathcal{S} \tag{12}$$

or equivalently

$$r_s^* = \min_{a \in L_s} H_{sa}(C_a - F_a^*), \quad \forall\, s \in S. \tag{13}$$

In some situations it may be reasonable to consider, in addition to (5) and (6), the constraint

$$r_s \leq R_s, \quad \forall\, s \in S \tag{14}$$

where R_s is given upper bound to the rate of session rate s. We may view R_s either as a limit on rate imposed by technological restrictions or as a maximum desired rate by session s. The problem of finding a fair allocation rate over the set defined by (5), (6), and (14) can be reduced to the problem considered earlier by introducing for each $s \in S$, an artificial link a_s traversed only by session s by setting the capacity C_{a_s} of that link equal to

$$C_{a_s} = R_s + \beta_s(R_s),$$

and by selecting the function g_{a_s} to be the identity. Then the constraint

$$\beta_s(r_s) \leq g_{a_s}(C_{a_s} - r_s) = C_{a_s} - r_s$$

becomes $r_s + \beta_s(r_s) \leq C_{a_s} = R_s + \beta_s(R_s)$ and is equivalent to (14).

3. THE ALGORITHM

Let $r^k = (\ldots, r_s^k, \ldots)$ be the rate vector obtained after k iterations and let $\{F_a^k\}$ be the corresponding set of total link flows. Assume that

$$0 \leq F_a^k < C_a, \qquad \forall\ a \in \mathcal{L}. \tag{15}$$

The new rate vector $r^{k+1} = (\ldots, r_s^{k+1}, \ldots)$ obtained at the (k+1)st iteration is given by

$$r_s^{k+1} = \min_{a \in \mathcal{L}_s} \{r_s^k + \gamma_a^k [H_{sa}(C_a - F_a^k) - r_s^k]\}, \quad \forall\ s \in \mathcal{S} \tag{16}$$

where γ_a^k is given by

$$\gamma_a^k = \frac{1}{1 + \sum_{s \in \mathcal{S}_a} H'_{sa}(C_a - F_a^k)} \tag{17}$$

and $H'_{sa}(\cdot)$ denotes the first derivative of $H_{sa}(\cdot)$. In a practical implementation of the algorithm the link flows F_a^k can either be measured (by taking a time average), or they can be mathematically computed as the sum of the session rates r_s^k, $s \in \mathcal{S}_a$. The session rates computed via (16), (17) will have to be translated into physical rates by software residing at the session origin nodes.

The following lemma shows among other things that property (15) is maintained by the algorithm at each iteration and therefore if the initial link flows F_a^o are within the link capacities C_a the same is true for all link flows generated by the algorithm.

Lemma 1: Let Assumptions A and B hold and assume that the initial rate vector r^o is such that

$$0 \leq r_s^o, \quad \forall\, s\epsilon S, \qquad 0 \leq F_a^o < C_a, \quad \forall\, a\epsilon L. \tag{18}$$

Then if $\{r^k\}$ is a rate sequence generated by the algorithm (16) with γ_a^k given by (17) we have for all k

$$0 \leq r_s^k, \quad \forall\, s\epsilon S, \qquad 0 \leq F_a^k < C_a, \quad \forall\, a\epsilon L. \tag{19}$$

Furthermore

$$F_a^k \leq \sum_{s\epsilon S_a} H_{sa}(C_a - F_a^k), \quad \forall\, a\epsilon L, \quad k \geq 1. \tag{20}$$

Proof: See Appendix A.

The idea behind the choice of expression (17) as well as the intuition behind Lemma 1 can be best explained by the use of Figure 1. Let the function $G_a(\cdot)\colon R^+ \to R^+$ be given by

$$G_a(F_a) = \sum_{s\epsilon S_a} H_{sa}(C_a - F_a) \tag{21}$$

The figure depicts the relations between F_a^k and F_a^{k+1}, as if the network consisted of the single link a. F_a^{k+1} is determined by intersection the tangent to the graph of $G_a(F_a)$ at the point $(F_a^k, G_a(F_a^k))$, with the line $y = F^a$. The reader can easily convince himself that $\limsup_{k\to\infty} F_a^k$ must lie in the area where

$$F_a \leq G_a(F_a)$$

which gives rise to the lemma. Figure 2 shows why just monotonicity of $G_a(\cdot)$ is not sufficient for the lemma to hold.

We can now state the main result of this paper.

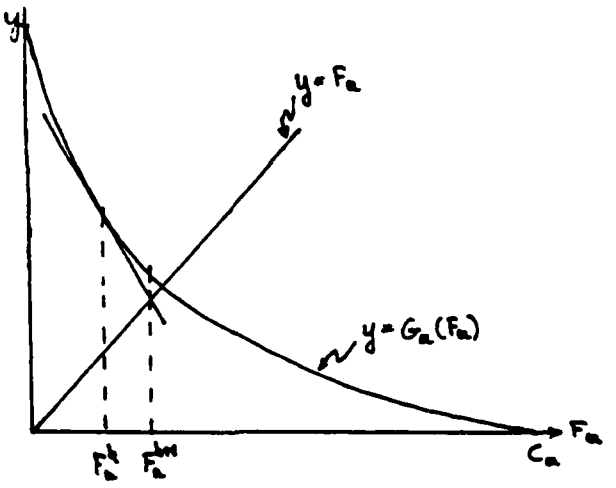

Figure 1

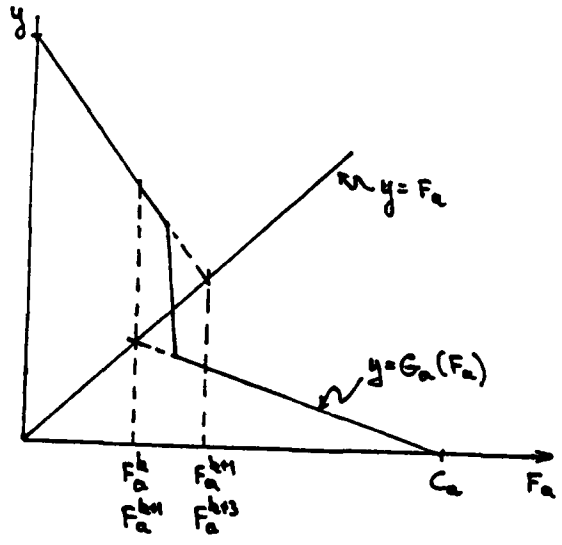

Figure 2

Proposition 2: Under the assumptions of Lemma 1 the sequence $\{r^k\}$ converges to the fair allocation rate.

Proof: See Appendix A.

A Variation of the Algorithm.

In iteration (16) we have assumed that updating of all the rates r_s takes place simultaneously. It is possible to consider other related algorithms whereby a single session rate r_s is updated using (16), then the flows F_a are updated to reflect the change in r_s, then another session rate is updated using (16) and so on until all the session rates are taken up cyclically in a fixed order. This one-session-at-a-time mode of operation is reminiscent of the Gauss-Seidel method for solving systems of equations and is perhaps better suited for distributed implementation. It is possible to show that all the results of this section hold for this modified algorithm as well.

A question of considerable interest is whether a totally asynchronous distributed version of algorithm (16) will work satisfactorily (compare with algorithms investigated in [5], [6]). In such an algorithm each session origin sends at arbitrary times along the session path a control packet containing the current rate of the session. As the packet travels to its destination the information needed to compute the right side of (16) is collected. (We assume here a single destination per session and that each link a on the session path maintains the current value of F_a as the sum of all currently assigned session rates r_s, $s \in S_a$, and

the form of the function $H_{sa}(\cdot)$ for each $s \epsilon S_a$). The destination returns the new rate to the session origin and the links along the session path. This type of algorithm is very attractive from the practical point of view since it does not require a session synchronization protocol. Its convergence properties however are as yet unclear and are currently under investigation. It is interesting to note that some of the algorithms investigated by simulation in [1] are of similar nature.

REFERENCES

[1] T. Bially, B. Gold, and S. Seneff, "A Technique for Adaptive Voice Flow Control in Integrated Packet Networks", IEEE Trans. Comm., Vol. COM-28, 1980, pp. 325-333.

[2] E. M. Gafni, "The Integration of Routing and Flow Control for Voice and Data in a Computer Communication Network", Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts, August 1982.

[3] J. M. Jaffe, "A Decentralized 'Optimal', Multiple-User, Flow Control Algorithm", Proceeding of Fifth Conference on Computer Communication (ICCC-80), Atlanta, Georgia, November 1980, pp. 839-844.

[4] H. Hayden, "Voice Flow Control in an Integrated Network", M.S. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts, June 1981.

[5] D. P. Bertsekas, "Distributed Asynchronous Computation of Fixed Points", Report LIDS-P-1135-A, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, August 1981, to appear in Mathematical Programming.

[6] D. P. Bertsekas, "Distributed Dynamic Programming", IEEE Trans. on Aut. Control, Vol. AC-27, 1982, pp. 610-616.

[7] F. H. Moss and A. Segall, "An Optimal Control Approach to Dynamic Routing in Networks", IEEE Trans. on Aut. Control, Vol. AC-27, 1982, pp. 329-339.

[8] B. Hajek and R. G. Ogier, "Optimal Dynamic Routing in Communication Networks with Continuous Traffic", Coordinated Science Lab. Report, Univ. of Illinois, Urbana, 1982.

APPENDIX A

Proof of Lemma 1:

It suffices to show (19)-(20) for k = 1. Consider the function $G_a: R^+ \to R^+$ defined by

$$G_a(F_a) = \begin{cases} \sum_{s \in S_a} H_{sa}(C_a - F_a) & \text{if } 0 \leq F_a \leq C_a \\ 0 & \text{if } F_a > C_a. \end{cases}$$

From (16) and (17) we have

$$r_s^1 = \min_{a \in L_s} \left\{ r_s^o + \frac{1}{1-G_a'(F_a^o)} \left[H_{sa}(C_a - F_a^o) - r_s^o \right] \right\}, \quad \forall\, s \in S$$

or

$$r_s^1 = \min_{a \in L_s} \left\{ \frac{H_{sa}(C_a - F_a^o) - r_s^o G_a'(F_a^o)}{1-G_a'(F_a^o)} \right\}, \quad \forall\, s \in S. \tag{A.1}$$

Since $G_a(\cdot)$ is monotonically nonincreasing we have $G_a'(F_a^o) \leq 0$, and since also $H_{sa}(C_a - F_a^o) \geq 0$ we obtain from the hypothesis $r_s^o \geq 0$ and (A.1)

$$r_s^1 \geq 0, \qquad \forall\, s \in S, \tag{A.2}$$

and therefore also

$$F_a^1 \geq 0, \qquad \forall\, a \in L. \tag{A.3}$$

From (A.1) we have

$$r_s^1 \leq \frac{H_{sa}(C_a - F_a) - r_s^o G_a'(F_a^o)}{1 - G_a'(F_a^o)} \quad , \quad \forall\ s\in S\ ,\ a\in L_s$$

and by adding over all $s\in S_a$ we obtain

$$F_s^1 \leq \frac{G_a(F_a^o) - F_a^o G_a'(F_a^o)}{1 - G_a'(F_a^o)} \quad , \quad \forall\ a\in L.$$

Since $1-G_a'(F_a^o) > 0$ we obtain from the inequality above

$$F_a^1 \leq G_a(F_a^o) + (F_a^1 - F_a^o)G_a'(F_a^o)$$

Since $G_a(\cdot)$ is convex the right side of this inequality is less or equal to $G_a(F_a^1)$ and we obtain

$$F_a^1 \leq G_a(F_a^1). \tag{A.4}$$

Since $G_a(\cdot)$ is monotonically nonincreasing and $G_a(F) = 0$ for $F \geq C_a$ we obtain from (A.4)

$$F_a^1 < C_a. \tag{A.5}$$

From (A.2)-(A.5) we obtain (19) and (20). Q.E.D.

Proof of Proposition 2:

Denote

$$r_s^* = \liminf_{k\to\infty} r_s^k \quad , \quad \forall\ s\in S.$$

Fix $s\epsilon S$ and consider a subsequence $\{r_s^k\}_{k\epsilon K_s}$ converging to r_s^*. We have from (16)

$$r_s^* = \lim_{\substack{k\to\infty \\ k\epsilon K_s}} \min_{a\epsilon L_s} \{r_s^{k-1} + \gamma_a^{k-1} [H_{sa}(C_a - F_a^{k-1}) - r_s^{k-1}]\} .$$

Since L_s consists of a finite number of links we may assume (by passing to a subsequence of K_s if necessary) that there exists a link a_s such that

$$r_s^* = \lim_{\substack{k\to K_s \\ k\to\infty}} \{r_s^{k-1} + \gamma_{a_s}^{k-1} [H_{sa_s}(C_{a_s} - F_{a_s}^{k-1}) - r_s^{k-1}]\} . \qquad (A.6)$$

Since $\{F_{a_s}^{k-1}\}_{k\epsilon K_s}$ is bounded above and below we may assume (by passing to a subsequence of K_s if necessary) that for some $\tilde{F}_{a_s}$

$$\lim_{\substack{k\to K_s \\ k\to\infty}} F_{a_s}^{k-1} = \tilde{F}_{a_s} .$$

Denote also

$$\tilde{\gamma}_{a_s} = \frac{1}{1-G'_{a_s}(\tilde{F}_{a_s})} = \lim_{\substack{k\epsilon K_s \\ k\to\infty}} \gamma_{a_s}^{k-1} .$$

We have from (A.6)

$$r_s^* \geq (1-\tilde{\gamma}_{a_s}) \liminf_{\substack{k\to K \\ k\to\infty}} r_s^{k-1} + \tilde{\gamma}_{a_s} H_{sa_s}(C_{a_s} - \tilde{F}_{a_s})$$

$$\geq (1-\tilde{\gamma}_{a_s}) r_s^* + \tilde{\gamma}_{a_s} H_{sa_s}(C_{a_s} - \limsup_{k\to\infty} F_{a_s}^k)$$

and finally

$$r_s^* \geq H_{sa_s}(C_{a_s} - \limsup_{k\to\infty} F_{a_s}^k) . \tag{A.7}$$

Since the choice of s was arbitrary we conclude that for every $s\varepsilon S$ there exists $a_s\varepsilon L_s$ such that (A.7) holds.

Let $a_1\varepsilon L$ be such that

$$a_1 = \arg\min_{a\varepsilon L} g_a(C_a - \limsup_{k\to\infty} F_a^k)$$

Using the monotonicity of β_s^{-1} we obtain

$$H_{sa_s}(C_{a_s} - \limsup_{k\to\infty} F_{a_s}^k) \geq H_{sa_1}(C_{a_1} - \limsup_{k\to\infty} F_{a_1}^k)$$

and therefore from (A.7)

$$r_s^* \geq H_{sa_1}(C_{a_1} - \limsup_{k\to\infty} F_{a_1}^k), \quad \forall\, s\varepsilon S_{a_1} . \tag{A.8}$$

Summing (A.8) over all $s\varepsilon S_{a_1}$ we obtain

$$\liminf_{k\to\infty} F_{a_1}^k \geq \sum_{s\varepsilon S_{a_1}} \liminf_{k\to\infty} r_s^k = \sum_{s\varepsilon S_{a_1}} r_s^*$$

$$\geq \sum_{s\varepsilon S_{a_1}} H_{sa_1}(C_{a_1} - \limsup_{k\to\infty} F_{a_1}^k)$$

On the other hand from (20) we have

$$\limsup_k F_{a_1}^k \leq \sum_{s\in S_{a_1}} H_{sa_1}(C_{a_1} - \limsup_k F_{a_1}^k).$$

It follows that the last two inequalities as well as (A.8) hold as equations, the entire sequence $\{F_{a_1}^k\}$ converges to $\sum_{s\in S_{a_1}} r_s^*$ while each sequence $\{r_s^k\}$, $s\in S_{a_1}$, converges to r_s^*.

Consider now a new network derived from the previous one by deleting link a_1, and all the sessions traversing it. We consider the algorithm executed in the same manner as before with the same initial rates for the remaining sessions but with the capacity of each link $a\in L$ replaced by

$$C_a - \sum_{s\in S_{a_1}} r_s^k.$$

This will result in the same rate sequence for the sessions $s\notin S_{a_1}$ as in the original algorithm. A trivial modification of the argument used to show (20) in Lemma 1 shows that we will have

$$\limsup_{k\to\infty} F_a^k \leq \sum_{s\in S_a} H_{sa}(C_a - \limsup_{k\to\infty} F_a^k) \quad \forall\ a\in L\ .$$

This relation can be used to repeat the argument given earlier in order to show the convergence of the sequences $\{r_s^k\}$ to r_s^* for all sessions $s\notin S_{a_1}$ traversing the link a_2 where

$$a_2 = \arg\min_{\substack{a\in L\\ a\neq a_1}} g_a(C_a - \limsup_{k\to\infty} F_a^k).$$

By repeating this procedure we will eventually exhaust all links thereby showing that each rate sequence $\{r_s^k\}$, $s\in S$ converges to r_s^*, each

each flow sequence $\{F_a^k\}$, $a \in L$ converges to $F_a^* = \sum_{s \in S_a} r_s^*$ and each stepsize sequence $\{\gamma_a^k\}$, $a \in L$ converges to

$$\gamma_a^* = \frac{1}{1 + \sum_{s \in S_a} H'_{sa}(C_a - F_a^*)} .$$

By taking limits in (16) we have for all $s \in S$

$$r_s^* = \lim_{k \to \infty} \min_{a \in L_s} \{r_s^k + \gamma_a^k [H_{sa}(C_a - F_a^k) - r_s^k]\} \qquad (A.9)$$

$$= \min_{a \in L_s} \lim_{k \to \infty} \{r_s^k + \gamma_a^k [H_{sa}(C_a - F_a^k) - r_s^k]\}$$

where the interchange of min and lim above is valid since all the sequences inside the braces converge and the number of elements of L_s is finite. From (A.9) we obtain for all $s \in S$

$$\min_{a \in L_s} \gamma_a^* [H_{sa}(C_a - F_a^*) - r_s^*] = 0$$

Since $\gamma_a^* > 0$ for all $a \in L_s$ we obtain

$$r_s^* = \min_{a \in L_s} H_{sa}(C_a - F_a^*) , \quad \forall \ s \in S.$$

The result now follows from Proposition 1 [cf. (13)]. Q.E.D.

Distribution List

Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12 Copies
Assistant Chief for Technology Office of Naval Research, Code 200 Arlington, Virginia 22217	1 Copy
Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	2 Copies
Office of Naval Research Branch Office, Boston 495 Summer Street Boston, Massachusetts 02210	1 Copy
Office of Naval Research Branch Office, Chicago 536 South Clark Street Chicago, Illinois 60605	1 Copy
Office of Naval Research Branch Office, Pasadena 1030 East Greet Street Pasadena, California 91106	1 Copy
Naval Research Laboratory Technical Information Division, Code 2627 Washington, D.C. 20375	6 Copies
Dr. A. L. Slafkosky Scientific Advisor Commandant of the Marine Corps (Code RD-1) Washington, D.C. 20380	1 Copy

Office of Naval Research — 1 Copy
Code 455
Arlington, Virginia 22217

Office of Naval Research — 1 Copy
Code 458
Arlington, Virginia 22217

Naval Electronics Laboratory Center — 1 Copy
Advanced Software Technology Division
Code 5200
San Diego, California 92152

Mr. E. H. Gleissner — 1 Copy
Naval Ship Research & Development Center
Computation and Mathematics Department
Bethesda, Maryland 20084

Captain Grace M. Hopper — 1 Copy
Naval Data Automation Command
Code 00H
Washington Navy Yard
Washington, DC 20374

Advanced Research Projects Agency — 1 Copy
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

Dr. Stuart L. Brodsky — 1 Copy
Office of Naval Research
Code 432
Arlington, Virginia 22217

Prof. Fouad A. Tobagi
Computer Systems Laboratory
Stanford Electronics Laboratories
Department of Electrical Engineering
Stanford University
Stanford, CA 94305